

SISTEM PAKAR MENDIAGNOSA PENYAKIT OSTEOPOROSIS MENGGUNAKAN METODE CERTAINTY FACTOR (CF)

Hardiana

Universitas cokroaminoto palopo

Abstrak

Penelitian ini bertujuan untuk merancang Sistem Pakar Mendiagnosa Penyakit *Osteoporosis* menggunakan Metode *Certainty Factor* (CF) yang didesain secara terstruktur menggunakan *Unified Modeling Language* (UML). Hasil penelitian menyimpulkan bahwa sistem pakar ini dapat memberikan kemudahan dalam mendiagnosa awal penyakit *osteoporosis* pada bagian punggung berdasarkan gejala yang diderita pasien, sehingga *user* dapat mengetahui apakah pasien terserang penyakit *osteoporosis* atau tidak. *Rule* yang tersimpan dalam *database* akan memberikan jawaban berupa identifikasi penyakit yang diderita.

Kata-kata kunci: Sistem Pakar, Certainty Factor, UML

I. PENDAHULUAN

1.1 Latar Belakang

Sistem bisnis dalam bidang IT pertama kali terjadi ketika CERN memungut bayaran dari para anggotanya untuk menanggulangi biaya operasionalnya. Pada 1992, mulai terbentuk komunitas internet dan diperkenalkannya istilah World Wide Web (www) oleh CERN. Tahun 1993, NSF membentuk InterNIC untuk menyediakan jasa pelayanan internet menyangkut direktori dan penyimpanan data serta database (oleh AT&T), jasa registrasi (oleh Network Solution Inc), dan jasa informasi (oleh General Atomics/CERFnet). Pada 1995, perusahaan umum mulai diperkenankan menjadi *provider* dengan membeli jaringan di *backbone*. Langkah ini memulai pengembangan teknologi informasi, khususnya internet dan penelitian-penelitian untuk mengembangkan sistem dan alat yang lebih canggih.

Sistem pakar adalah program *artificial intelligence* yang menggabungkan pangkalan pengetahuan (*knowledge base*) dengan sistem inferensi. Perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam memecahkan suatu permasalahan.

Aplikasi sistem pakar adalah sistem yang meniru kepakaran (keahlian) seseorang dalam bidang tertentu dalam menyelesaikan suatu permasalahan. Sering kali sistem pakar disebut sistem berbasis pengetahuan (*knowledge based system*).

Sistem pakar diagnosis biasanya menggunakan deskripsi keadaan,

karakteristik tingkah laku, atau pengetahuan tentang pembuatan komponen. Sistem pakar diagnosis biasanya menggunakan deskripsi keadaan, karakteristik tingkah laku, atau pengetahuan tentang pembuatan komponen sehingga dapat menentukan kemungkinan kerusakan pada sistem. Contohnya adalah menentukan penyakit dari gejala-gejala yang terlihat pada pasien. Menentukan lokasi kesalahan pada rangkaian listrik, atau mencari komponen yang rusak dalam sistem pendingin reaktor nukir. Biasanya sistem pakar diagnosis menggunakan pohon keputusan (*decision tree*) sebagai representasi pengetahuannya.

Berdasarkan uraian diatas, maka saya mencoba mencermati lebih lanjut mengenai masalah tersebut sehingga menjadikannya suatu masalah yang perlu dipaparkan dalam laporan dengan judul “Sistem Pakar Mendiagnosa Penyakit Osteoporosis Menggunakan Metode Certainty Factor (CF)”.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang sistem pakar mendiagnosa penyakit *osteoporosis* dengan menggunakan metode *Certainty Factor* (CF)?
2. Sistem pakar bisa dimanfaatkan untuk mendeteksi penyakit *osteoprosis* yang diimplementasikan ke dalam bentuk aplikasi?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini diantaranya:

1. Merancang sistem pakar sistem pakar mendiagnosa penyakit *osteoporosis*.
2. Menggunakan aplikasi *microsoft visual basic 6.0* dalam membuat sistem pakar dan UML dengan *database Access*.
3. Metode yang digunakan adalah metode *Certainty Factor*.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Untuk merancang sistem pakar mendiagnosa penyakit *osteoporosis*.
2. Untuk mengimplementasikan sistem pakar mendeteksi penyakit *osteoporosis* pada manusia dalam bentuk *software* aplikasi sebagai sebuah alat bantu dalam memberikan pengetahuan dasar tentang penyakit *osteoporosis*.

1.5 Manfaat Penelitian

Manfaat dari sistem pakar ini adalah:

1. Manfaat terhadap penulis
Dapat meneliti dan memahami cara kerja dari sebuah sistem pakar sehingga dapat mendiagnosa penyakit *osteoporosis* pada manusia
2. Manfaat bagi instansi:
 - a. Menjadi salah satu bahan referensi dan informasi bagi pihak RSUD Sawerigading Palopo
 - b. Menjadikan acuan lebih maju sesuai dengan perkembangan teknologi.

II. TINJAUAN PUSTAKA

2.1 Konsep dasar aplikasi

Pengertian program aplikasi secara umum merupakan penerapan, menyimpan sesuatu hal, data, permasalahan pekerjaan kedalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan.

Menurut Jogiyanto (2000:112), "Program aplikasi merupakan ekspresi, pernyataan kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan

bahasa pemrograman, sehingga dapat dieksekusi oleh komputer".

Secara historis, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. App adalah *software* yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang paralel dengan yang terjadi antar sistem operasi yang dimunculkan.

Dari beberapa definisi tentang aplikasi dapat disimpulkan bahwa aplikasi adalah sebuah program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus. Klasifikasi aplikasi dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi terbagung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, dengan dokumentasi terbagung yang dirancang untuk jenis masalah tertentu.

2.2 Konsep dasar perancangan aplikasi

Menurut Jogiyanto (2001:24), Konsep merancang multimedia merupakan aplikasi multimedia yang akan dibuat. Untuk dapat merancang konsep dalam membuat aplikasi multimedia dibutuhkan kreatifitas. Kreatifitas adalah kemampuan untuk menyajikan gagasan atau ide baru. Sedangkan inovasi merupakan aplikasidari gagasan atau ide baru tersebut.

Multimedia yang telah ada dianggap belum sesuai dengan lingkungan yang dituju walaupun isinya telah sesuai dengan lingkungan. Misalnya multimedia ditujukan ke kalangan kawula muda namun multimedia yang ada hanya sesuai untuk kalangan orang tua (dari sisi desain tampilan, bahasa dll) pembesaran (*maksimasi*).

2.3 Unified Modeling Language (UML)

Menurut STOA (2008), *Unified Modeling Language* merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemograman yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan

oleh banyak perusahaan raksasa seperti IBM, Microsoft dan sebagainya .

1. Pengenalan diagram-diagram UML











Banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML yang di gunakan dalam penelitian ini adalah:

1. Diagram *use case* (*use case diagram*)
2. Diagram aktivitas (*activity diagram*)
3. Diagram sekuensial (*sequence diagram*)
4. Diagram kelas (*class diagram*)

a. Diagram Use Case

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. Simbol *use case* diagram dapat dilihat pada tabel 1 berikut ini:

Tabel 1. Simbol Use Case Diagram






No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : Hans-Erik Erikson dan Magnus Penker. *Busines Modeling With UML*, John Wilcy & Son, Inc, New York, USA. 2000

b. Diagram aktivitas

Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagrama aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*) dalam *use case*. Simbol *aktivity diagram* dapat dilihat pada tabel 2 berikut ini:

Tabel 2. Simbol Activity Diagram




No	Gambar	Nama	Keterangan
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan dihancurkan
5		Fork Node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : Hans-Erik Erikson dan Magnus Penker. *Busines Modeling With UML*, John Wilcy & Son, Inc, New York, USA. 2000

c. Diagram sekuence

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel 3 berikut ini:

Tabel 3. Simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1		LifeLine	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi








Sumber : Hans-Erik Erikson dan Magnus Penker. *Busines Modeling With*

UML, John Wilcy & Son, Inc, New York, USA. 2000

d. Diagram Kelas (*Class Diagram*)

Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Ia memberikan gambaran sistem secara statis. Biasanya, dibuat beberapa diagram kelas untuk satu sistem. Simbol yang digunakan dapat dilihat pada tabel 4 berikut ini:

Tabel 4. Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber : Hans-Erik Erikson dan Magnus Penker. *Busines Modeling With UML*, John Wilcy & Son, Inc, New York, USA. 2000

2.4 Pengertian sistem pakar

Sistem pakar adalah program *artificial intelligence* yang menggabungkan pangkalan pengetahuan (*knowledge base*) dengan sistem inferensi. Perangkat lunak komputer yang memiliki basis pengetahuan untuk *domain* tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam memecahkan suatu permasalahan. Sistem pakar adalah sebuah teknik inovatif baru dalam menangkap dan memadukan pengetahuan, kekuatan sistem pakar terletak pada kemampuannya memecahkan masalah-masalah praktis pada saat seorang pakar berhalangan.

Beberapa definisi yang ada untuk sistem pakar (*expert system*):

- 1) Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar (Kusrini; 2008).
- 2) Sistem pakar adalah salah satu cabang dari kecerdasan buatan yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar (Arhami, Muhammad: 2005).

Ciri-ciri sistem pakar

Pengetahuan sistem pakar terdapat ciri-ciri sebagai berikut:

- 1) Sifatnya terbatas pada domain keahlian tertentu.
- 2) Dapat memberikan penalaran untuk data-data yang tidak pasti.
- 3) Dapat mengemukakan serangkaian alasan yang diberikan dengan cara yang dapat dipahami.

2.5 Osteoporosis

Osteoporosis adalah berkurangnya kepadatan tulang yang *progresif*, sehingga tulang menjadi rapuh dan mudah patah. Tulang terdiri dari mineral-mineral seperti *kalsium* dan *fosfat*, sehingga tulang menjadi keras dan padat. Untuk mempertahankan kepadatan tulang, tubuh memerlukan persediaan kalsium dan mineral lainnya yang memadai dan harus menghasilkan hormon dalam jumlah yang mencukupi (hormon *paratiroid*, hormon pertumbuhan, *kalsitonin*, *estrogen* pada wanita dan *testosteron* pada pria). Juga persediaan vitamin D yang adekuat, yang diperlukan untuk menyerap kalsium dari makanan dan memasukkan ke dalam tulang.

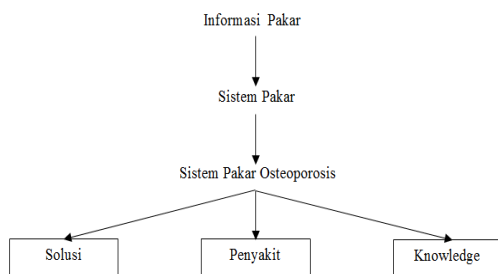
2.6 Metode *Certainty factor* (CF)

Model *certainty factor* adalah metode untuk mengelola ketidakpastian dalam sistem berdasarkan aturan. Shortliffe dan Buchanan (1975) mengembangkan model CF di pertengahan 1970-an untuk MYCIN, sistem pakar untuk diagnosis dan pengobatan meningitis dan infeksi darah. Sejak itu, model CF telah menjadi pendekatan standar untuk manajemen ketidakpastian dalam sistem berdasarkan aturan.

III. ANALISIS PERANCANGAN BERORIENTASI OBJEK

3.1 Perancangan sistem

Kebutuhan fungsional yang ada di sistem pakar adalah informasi yang dapat dilihat yaitu sistem pakar mendiagnosa penyakit *osteoporosis* yang terdiri dari beberapa alternatif yakni pasien, solusi, penyakit dan *knowledge* (basis pengetahuan). Hirarki sistem pakar mendiagnosa penyakit *osteoporosis* dapat dilihat pada gambar 3 berikut ini:



Gambar 3. Hirarki sistem pakar *osteoporosis*

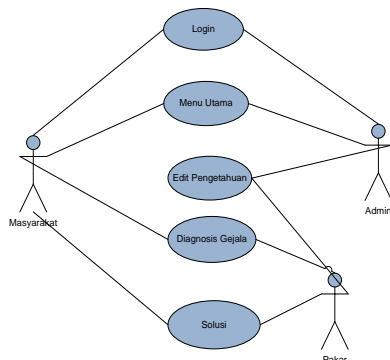
1. Sistem yang diusulkan

Sistem yang diusulkan yaitu *admin* dan user masuk ke menu utama kemudian login dan melakukan diagnosa dengan cara memberikan pertanyaan dan pasien memberi jawaban kemudian *admin* memilih pilihan jawaban yang ada dalam aplikasi, setelah itu melakukan proses untuk mendapatkan hasil diagnosa dan memberikan hasil serta solusi kepada pasien.

2. Perancangan proses

1. Use Case Diagram

Rancangan *use case* dapat dilihat pada gambar berikut yaitu *admin* melihat tampilan utama aplikasi dan mengimput kriteria dan *alternative* sesuai dengan aplikasi dan hasilnya berupa keputusan. Untuk lebih jelasnya dapat di lihat pada gambar 4 berikut:



Gambar 4. Use Case Diagram

2. Class – Responsibility – Collaboration (CRC)

Rancangan model CRC digunakan sebagai dokumentasi tanggung jawab dan kolaborasi dari masing- masing *class*. Berikut ini dapat dilihat CRC dari Sistem Pakar Mendiagnosa Penyakit Osteoporosis menggunakan Metode CF. Untuk lebih jelasnya dapat di lihat pada tabel berikut:

Tabel 5. CRC Class Login

Class	: Login
Responsibilities :	Collaborator :
a. Get form login	
b. Login admin	Login admin
c. Service login (username & password sukses/gagal)	Login masyarakat
d. Keluar	

Tabel 6. CRC Class Solusi

Class	: Solusi
Responsibilities :	Collaborator :
b. GetSolusi	Solusi
c. Lihat Solusi	

Tabel 7. CRC Class Knowledge

Class	: Knowledge
Responsibilities :	Collaborator :
a. Getpertanyaan	Pertanyaan
b. GetFaktaYa	Fakta Ya
c. GetFaktaTidak	Fakta tidak
d. GetYa	Ya
e. GetTidak	Tidak
f. Lihat pertanyaan	
g. Lihat fakta ya	
h. Lihat fakta tidak	

Tabel 8. CRC Class Penyakit

Class	: Penyakit
Responsibilities :	Collaborator :
a. Getpenyakit	Penyakit
b. GetJenis	
c. Lihat penyakit	Jenis
d. Lihat jenis	

Tabel 9. CRC Class Pasien

Class	: Pasien
Responsibilities :	Collaborator :
a. GetPasien	Pasien
b. Lihat	Pasien

IV. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi sistem

Dalam pembuatan aplikasi ini tentunya diperlukan kebutuhan perangkat lunak dan perangkat keras. Maka dari itu, kebutuhan-kebutuhan tersebut diantaranya:

1. Implementasi perangkat keras dan perangkat lunak

Adapun perangkat lunak yang digunakan pada saat implementasi yaitu :

1. Sistem Operasi menggunakan Windows 7.
2. Basis data menggunakan Microsoft Access 2007.

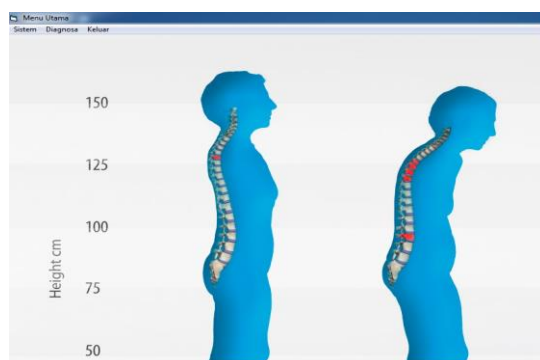
3. Aplikasi yang digunakan Visual basic.

Adapun perangkat keras yang digunakan pada saat implementasi yaitu :

- a. Processor Intel i3 2.27 GHz
- b. Harddisk 250 GB
- c. RAM 1 GB
- d. Perangkat standar Input dan output.

2. Implementasi menu utama

Tampilan menu utama dapat dilihat pada gambar 5 berikut:



Gambar 5. Implementasi menu utama

3. Implementasi admin

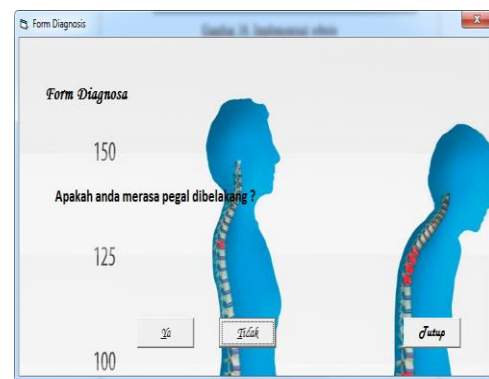
Tampilan implementasi admin dapat dilihat pada gambar 6 berikut:



Gambar 6. Implementasi admin

4. Implementasi user

Tampilan implementasi user dapat dilihat pada gambar 7 berikut:



Gambar 7. Implementasi user

Hasil pengujian Sistem Pakar Mendiagnosa Penyakit Osteoporosis dapat dilihat pada tabel 15 berikut:

Tabel 10. Hasil pengujian program

N o.	Nama Bagan Alir Program	Jumlah CC	Independent Path	Jumlah Region
1.	Menu Utama	8	8	8
2.	Login	3	3	3
3.	Menu Gejala Edit	6	6	6
4.	Pengetahuan	3	3	3
5.	Pertanyaan	6	6	6
6.	Solusi	4	4	4
7.	Diagnosa Gejala	6	6	6
	Total	36	36	36

Berdasarkan tabel hasil pengujian program diatas, ternyata jumlah *Region*, *Cyclomatic Complexity* (CC) dan *Independent Path* adalah sama besar sehingga dapat ditarik kesimpulan bahwa program telah bebas dari kesalahan program tersebut. Dengan kata lain bahwa Sistem Pakar Mendiagnosa Penyakit Osteoporosis yang dibuat telah layak digunakan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Hasil perancangan sistem pakar mendiagnosa penyakit *osteoporosis*.
2. Hasil pengujian sistem menggunakan *White Box* telah menunjukkan bahwa pembuatan sistem telah bebas dari kesalahan pada program tersebut.
3. Hasil implementasi sistem pakar mendiagnosa penyakit *osteoporosis*.

5.2 Saran

Setelah melakukan penelitian RSUD Sawerigading Kota Palopo, maka penulis mengajukan saran sebagai berikut:

1. Melihat hasil penelitian dimana penggunaan sistem ini mudah-mudahan akan dapat memudahkan masyarakat dalam mengetahui gejala-gejala dari Penyakit *Osteoporosis*, maka diharuskan kedepannya Sistem Pakar Mendiagnosa Penyakit *Osteoporosis* bisa diterapkan.
2. Dengan diterapkannya Sistem Pakar Mendiagnosa Penyakit *Osteoporosis*, maka akan dijadikan perbandingan antara sistem yang lama dengan sistem yang baru, supaya sistem ini dapat diterima dan digunakan.

Demikian kesimpulan dan saran yang penulis bisa ambil dan diharapkan kiranya apa yang penulis lakukan ini dapat menjadi pertimbangan kepada RSUD Sawerigading Kota Palopo.

DAFTAR PUSTAKA

- Arhami, M. 2005. Konsep Dasar Sistem Pakar. ANDI: Yogyakarta.
- Kusrini. 2006. Sistem Pakar Teori dan Aplikasi. Andi: Yogyakarta.

- Kusrini. 2008. Aplikasi Sistem Pakar. Andi: Yogyakarta.
- Kusrini. 2006. Seminar ilmiah, Kuantifikasi Pertanyaan Untuk Mendapatkan *Certainty Factor* Pengguna Pada Aplikasi Sistem Pakar Untuk *Diagnosis* Penyakit. Andi: Yogyakarta.
- STOA. 2008. Unified Modeling Language. Raja Grafindo: Jakarta.
- Sunyoto. 2007. Pemrograman Database dengan Visual Basic dan Microsoft SQL. Tim penerbit Andi: Yogyakarta.
- Tim Divisi Penelitian dan Pengembangan MADCOMS, 2008. Microsoft Access 2007 untuk pemula. Andi: Madiun.
- Tim Penerbit Andi, 2003. Pengembangan Sistem Pakar Menggunakan Visual Basic. Tim penerbit Andi: Yogyakarta.
- Turban dkk. 2005. Decision Support System and Intelligent System (Sistem Pendukung Keputusan dan Sistem Cerdas). Andi: Yogyakarta.